```
     (FILE 'HOME' ENTERED AT 14:49:28 ON 26 MAY 2000)

     FILE 'USPATFULL' ENTERED AT 14:49:35 ON 26 MAY 2000
L1              8 S HEADER? (P) PARS? (P) HASH###
L2             44 S SUMMARY (4A) HEADER?
L3             94 S HASH### (6A) HEADER?
L4             63 S L3 AND PROTOCOL?
L5            107 S L2 OR L4
L6            526 S BYPASS? (P) PROTOCOL?
L7              0 S L6 AND L5
L8              0 S L3 AND L6
L9              3 S L3/AB
L10         86074 S CONTROL? (4A) BLOCK?
L11             2 S L3 (P) L10
L12             8 S L2 AND L10

=> d l2 pn,ab,kwic


L2   ANSWER 1 OF 44  USPATFULL
PI     US 6065058  20000516
AB     A push-based filtering of objects in a client-server hierarchy based on
       usage information. A method of annotating a push object with meta
       information on its content and/or urgency is also described. Objects can
       be staged at the server(s) to provide fast access when the filtered
       object is later requested. The PICS protocol may be used to communicate
       various types of information: e.g., by the content provider or a higher
       level proxy to annotate the object, including an urgency, a summary or
       title, a group classification, and/or an identity of the push; to convey
       usage or preference information on pushed objects up the hierarchy,
       including usage information and user preferences based on object group
       classifications; and to convey a staging status of each staged object
       down the hierarchy to improve caching efficiency. An object may include
       a content hierarchy such as a title, a summary and the full content. The
       filtering process can factor in not only which next (lower) level nodes
       will receive the push, but also the content level each node will
       receive. The push filtering decision can be based on aggregate usage
       information at the lower level proxy or client nodes. A staging decision
       can be based on the filtering decision, the successful completion of the
       push to the selected lower level proxies/clients, object usage
       information and/or a staging decision on other nodes in the hierarchy.
       An efficient means to purge a staged object is also described.
DETD   . . . object O will be pushed to all next lower level nodes with P(i)
       equal to 1. In step 1140, a summary header of the
       object O will be pushed to all next lower level nodes with P(i) equal to
       0.5. In step. . .


=> d l2 1-44 pn,ab,kwic
```

```
     (FILE 'HOME' ENTERED AT 14:49:28 ON 26 MAY 2000)

     FILE 'USPATFULL' ENTERED AT 14:49:35 ON 26 MAY 2000
L1             8 S HEADER? (P) PARS? (P) HASH###
L2            44 S SUMMARY (4A) HEADER?
L3            94 S HASH### (6A) HEADER?
L4            63 S L3 AND PROTOCOL?
L5           107 S L2 OR L4
L6           526 S BYPASS? (P) PROTOCOL?
L7             0 S L6 AND L5
L8             0 S L3 AND L6
L9             3 S L3/AB

=> s control? (4a) block?

       1480131 CONTROL?
        848598 BLOCK?
L10      86074 CONTROL? (4A) BLOCK?

=> s l3 (p) l10

L11            2 L3 (P) L10
```

L1 ANSWER 2 OF 8 USPATFULL

PI US 6061368 20000509

AB Custom circuitry for an adaptive, RAM-based, hardware routing engine conforms bit selection dynamically to information made available by a self-optimizing data hashing algorithm. Circuitry includes a staggered multiplexor array for selecting bits from only the most distinctive bit positions of an inbound identifier to effectuate associative comparisons at approximately CAM speeds under perfect or near perfect hash conditions. The custom circuitry includes means for monitoring routing performance and instructing a processing element to correct the hashing algorithm when performance has sufficiently deteriorated. The custom circuitry also has an extended recursive look-up capability.

DETD Referring to FIG. 2, routing engine 100 is shown in greater detail. Engine 100 includes packet **parsing** unit 210, pseudo CAM logic 220, **header** cache 230 and packet assembly 240. Packet **parsing** unit 210 **parses** inbound packets based on protocol and extracts identifier sets for use in associative comparisons performed within engine 100. In a. . . a 32-bit IPX destination network identifier and a 48-bit IPX destination node identifier, for a total of eighty bits. Packet **parsing** unit 210 transmits each extracted identifier set to pseudo CAM logic 220 along with an associated look-up request, and transmits. . . contents to packet reassembly 240. Pseudo CAM logic 220 is operative in one or more look-up modes, in conjunction with **hash** RAM 120, to perform associative comparisons in **hash** RAM 120 on components from each identifier set received from packet **parsing** unit 210 and to issue, for any matching entry, a **header** index for retrieving routing information from **header** cache 230. The routing information retrieved from **header** cache 230 may then be transmitted to packet reassembly 240 for encoding in the **header** of an outbound packet associated with the inbound packet from which the identifier set was extracted. Packet reassembly 240 reassembles the outbound packet using the routing information and the remaining packet contents received from packet **parsing** unit 210 and transmits the outbound packet on output pin DATA.sub.-- OUT. Packet **parsing** unit 210, **header** cache 230 and packet assembly 240 may be implemented in custom circuitry using elements and techniques known to the art.

DETD Referring now to FIG. 8b, in a more preferred embodiment, global entry sets are configured in **hash** RAM 120 for each IP and IPX identifier set to enable pseudo CAM logic 220 to perform single-mode global associative comparisons. In the more preferred embodiment, entry sets in **hash** RAM 120 include global IP entry set 850 and global IPX entry set 860. Each entry in global entry sets. . . Operation proceeds generally as in the preferred embodiment except that the match command from mode registers 410 causes identifier set **parsing** unit 430 to transmit to match control 340 a global match identifier which includes all identifiers within the identifier set to be matched. The match information retrieved from **hash** RAM 120 is compared with the global match identifier such that further look-up modes are obviated. If a global match is found, match control 340 simply transmits the **header** index associated with the matching entry to **header** cache 230 for generating an outbound packet

```
L8    ANSWER 1 OF 1  USPATFULL
PI      US 5987432  19991116
AB      A central ticker plant system for distributing financial market data
        that receives ticker feed data from many exchanges throughout the world,
        processes and formats the received data and then distributes or
        broadcasts the data to regional customers in the form of securities
        transactional data denoting the security identity and related
        transactional data. The central ticker plant system is fault-tolerant
        because of novel hardware redundancy and multi-thread software
        processing architecture and operates continuously during hardware and
        software maintenance and repair, ensuring that every financial market
        data message received from the exchanges is included within 500
        milliseconds in broadcast output.


=> d 19 pn,ab


L9    ANSWER 1 OF 1  USPATFULL
PI      US 5802278  19980901
AB      A high performance scalable networking bridge/router system is based on
        a backbone communication medium and message passing process which
        interconnects a plurality of input/output modules. The input/output
        modules vary in complexity from a simple network interface device having
        no switching or routing resources on board, to a fully functional
        bridge/router system. Also, in between these two extremes input/output
        modules which support distributed protocol processing are supported. A
        central internetworking engine includes a shared memory resource coupled
        to the backbone. The architecture includes a physical layer
        communication system for transferring control messages and data packets
        across the backbone, a logical layer interprocessor messaging system
        which operates over the physical layer across the backbone for
        communication between intelligent input/out modules, and between such
        modules in the central internetworking engine, and distributed protocol
        modules which are supported on intelligent input/output modules, and
        communicate using the logical interprocessor messaging system with the
        central internetworking resources.


=> d kwic 16 1-23 pn,ab,kwic


L6    ANSWER 1 OF 23  USPATFULL
DETD    .  .  .  disabled during runtime through the MULTISPAN LOAD SHARING
        command, which toggles this mode. When a packet is sent from the
      protocol stack 100 to the primary NIC 124 (the board
        which is known to the protocol stack), the MULTISPAN
        system intercepts the request and selects the next active board from the
        group on which the packet could.  .  .  gets a turn to send packets. If
        a selected board in the bound group is marked "DISABLED", the MULTISPAN
        system bypasses that board and selects the next active board
        in the group. In another embodiment, the algorithm used makes a
        calculation.  .  .
PI      US 6052733  20000418
AB      A method is described for providing fault tolerance within a computer
        system. The method allows multiple network interface cards to reside
        within the same computer system. If a primary network interface card
```

fails, a secondary network interface card automatically begins managing the network communications. In addition, a method [illegible] load-sharing data transmissions between each network interface card installed in a server computer is described.

DETD . . . disabled during runtime through the MULTISPAN LOAD SHARING command, which toggles this mode. When a packet is sent from the **protocol stack** 100 to the primary NIC 124 (the board which is known to the **protocol stack**), the MULTISPAN system intercepts the request and selects the next active board from the group on which the packet could. . . gets a turn to send packets. If a selected board in the bound group is marked "DISABLED", the MULTISPAN system **bypasses** that board and selects the next active board in the group. In another embodiment, the algorithm used makes a calculation. . .

L6 ANSWER 2 OF 23 USPATFULL

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872 to. . .

PI US 6026379 20000215

AB An architecture is disclosed allowing a server to communicate bidirectionally with a gateway over a first communication link, over which service requests are initiated by the server. In response to a transaction received from a host legacy system at the gateway, the gateway parses one or more transaction response values from the host message, maps the one or more transaction response values to a canonical response code, and stores the canonical response code in a transaction log. According to a broad aspect of a preferred embodiment of the invention, communication networks that employ transactions between applications must effectively manage transactions that flow over the network. In addition, networking systems must also detect counterfeit transactions, especially, when the networking systems are utilized for financial transactions. An active, on-line database is utilized as a transaction log to track original requests, valid retrys and detect fradulant transactions. The transaction log serves as a memory cache where the received host response is returned to a valid retry transaction should the original response fail to reach a server because of a communications problem.

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872 to. . .

L6 ANSWER 3 OF 23 USPATFULL

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then,

at function block 1870 the gateway engine processes SET and Host
specific code including gateway administration extensions 1872 that
**bypass** the normal processing and flow directly from the merchant
and consumer server 1820 to the gateway administration extensions 1872
to. . .

PI US 6002767 19991214

AB Secure transmission of data is provided between a plurality of computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from a customer computer system to a merchant computer system, and for the further secure transmission of payment information regarding a payment instrument from the merchant computer system to a payment gateway computer system. The payment gateway system evaluates the payment information and returns a level of authorization of credit via a secure transmission to the merchant which is communicated to the customer by the merchant. The merchant can then determine whether to accept the payment instrument tendered or deny credit and require another payment instrument. An architecture that provides support for additional message types that are value-added extensions to the SET protocol is provided by a preferred embodiment of the invention. A server communicating bidirectionally with a gateway is disclosed. The server communicates to the gateway over a first communication link, over which all service requests are initiated by the server. The gateway uses a second communication link to send service signals to the server. In response to the service signals, the server initiates transactions to the gateway or presents information on an a display device.

DETD . . . change over time as more functionality is put into SET) or SET
extensions. Set compliant messages are processed via the
**protocol** statck library 1862, while SET extensions are processed
via the **protocol stack** entension library 1864. Then,
at function block 1870 the gateway engine processes SET and Host
specific code including gateway administration extensions 1872 that
**bypass** the normal processing and flow directly from the merchant
and consumer server 1820 to the gateway administration extensions 1872
to. . .

L6 ANSWER 4 OF 23 USPATFULL

DETD . . . change over time as more functionality is put into SET) or SET
extensions. Set compliant messages are processed via the
**protocol** statck library 1862, while SET extensions are processed
via the **protocol stack** entension library 1864. Then,
at function block 1870 the gateway engine processes SET and Host
specific code including gateway administration extensions 1872 that
**bypass** the normal processing and flow directly from the merchant
and consumer server 1820 to the gateway administration extensions 1872
to. . .

PI US 5996076 19991130

AB Secure transmission of data is provided between a plurality of computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from a party in communication with a first application resident on a first computer which is in communication with a second computer with a certification authority application resident thereon. The second computer is in communication with a third computer utilizing an administrative function resident thereon. The first, second and third computers are connected by a network, such as the Internet. A name-value pair for certification processing is created on said first computer and transmitted to an administrative function on the third computer. Then, the name-value pair is routed to the appropriate certification authority on the second computer. The administrative function also transmits other certification information from said administrative function to said certification authority on the second computer. Until, finally, a certificate is created comprising the name-value pair and the other certification information on the second computer. The certificate is utilized for authenticating identity of the party.

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872 to. . .

L6 ANSWER 5 OF 23 USPATFULL

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872 to. . .

PI US 5987132 19991116

AB An architecture that provides a server that communicates bidirectionally with a gateway over a first communication link, over which service requests flow to the server for one or more merchants and/or consumers is disclosed. Service requests are associated with a particular merchant based on storefront visited by a consumer or credentials presented by a merchant. Service requests result in merchant specific transactions that are transmitted to the gateway for further processing on existing host applications. By presenting the appropriate credentials, the merchant could utilize any other computer attached to the Internet utilizing a SSL or SET protocol to query the vPOS system remotely and obtain capture information, payment administration information, inventory control information, audit information and process customer satisfaction information.

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872 to. . .

L6 ANSWER 6 OF 23 USPATFULL

DETD . . . change over time as more functionality is put into SET) or SET extensions. Set compliant messages are processed via the **protocol** statck library 1862, while SET extensions are processed via the **protocol stack** entension library 1864. Then, at function block 1870 the gateway engine processes SET and Host specific code including gateway administration extensions 1872 that **bypass** the normal processing and flow directly from the merchant and consumer server 1820 to the gateway administration extensions 1872

PI      US 5991299  19991123

AB      A method and apparatus is disclosed for translating data link layer and network layer frame headers at speeds approximating the reception rate of frames on respective communication links. High-speed header translation is achieved via the use of a dedicated microsequencer which identifies the receive frame encapsulation type and the transmit frame encapsulation type and based on such identification, selects a processing routine which is then executed to translate the frame header. The microsequencer is employed to control the movement of control information and frame header and payload information from an input FIFO, through the dedicated header processor, and to an output FIFO. The headers of the respective frames are translated within the dedicated header processor to facilitate header translation at high speeds. Via use of the presently disclosed header translation apparatus, layer 2 and layer 3 header translations, as well as other header translation functions may be rapidly performed.

SUMM    . . . to permit all header translations to be controlled via microcode. In this manner the transmit header processor may accommodate new **protocols** or **bypass** broken hardware via the use of such microcode.

DETD    Elements in the motherboard 12 and interface modules 14 are responsible for data unit reception and transmission, **parsing** of data link and network layer **headers** within received frames, look-up of source and destination Media Access Control ("MAC") and network layer addresses and for making forwarding. . .

DETD    . . . State Machine ("RSM") 40, for receiving frames at respective ports 18 of the respective network interface module 10, a Receive **Header** Processor ("RHP") 46 for **parsing** **headers** of received frames, Buffer RAM 22 for storing received frames, a Receive Segment Processor (RSEG) 50 for dividing the received. . .

DETD    . . . of greater than or equal to 1500 in that location indicates a type field. Based upon the results from the **parsing** of the layer 2 **header** a decision is made as to whether to employ a Type Table or an LLC table to identify a Protocol. . .

CLM     What is claimed is:
. . . frame having a transmit frame header prior to transmission of said frame, said apparatus comprising; a first controller operable to **parse** said received frame **header** to derive a received destination address from said header and to identify the protocol of said received frame; cache apparatus. . .

L3   ANSWER 2 OF 3  USPATFULL

PI      US 5774660  19980630

AB      A multi-node server transmits world-wide-web pages to network-based browser clients. A load balancer receives all requests from clients because they use a virtual address for the entire site. The load balancer makes a connection with the client and waits for the URL from the client. The URL specifies the requested resource. The load balancer waits to perform load balancing until after the location of the requested resource is known. The connection and URL request are passed from the load balancer to a second node having the requested resource. The load balancer re-plays the initial connection packet sequence to the second node, but modifies the address to that for the second node. The network software is modified to generate the physical network address of the second node, [illegible]t then changes the destination [illegible]ress back to the virtual address. The second node transmits the requested resource directly to the client, with the virtual address as its source. Since all requests are first received by the load balancer which determines the physical location of the requested resource, nodes may contain different resources. The entire contents of the web site is not mirrored onto all nodes. Network bottlenecks are avoided since the nodes transmit the large files back to the client directly, bypassing the load balancer. Client browsers can cache the virtual address, even though different nodes with different physical addresses service requests.

DETD . . . request 102 in a PUSH packet. A PUSH packet is identified by a PUSH flag being set in the TCP **header**. The load balancer **parses** the URL to get the file or resource name. Based on the requested resource, and the location of each resource. . .

DETD . . . IP layer to the link layer. Step 336 of FIG. 16 detects that the local server is the destination and **bypasses** steps 338, 340 so that the **protocol** is left as IXP.

L3 ANSWER 13 OF 36 USPATFULL
PI US 5848410 19981208
AB To improve access to documents in a large database, such as the World Wide Web of the Internet or a group intranet, a continuously updated computer organization and display system, and a method for such an organization and display system, is provided to quickly locate desired documents without generating references to undesired documents, and to quickly allow a user to determine if any documents of interest are available. In one embodiment, such organization and display system includes data structures for storing and processing information extracted from the header lines of web pages in file systems chosen by a user. Linked lists are created in such data structures to allow rapid construction and display of an alphabetical index of keywords from such header lines, each keyword having associated with it a title extracted from the same web-page header. The alphabetical index by keywords may be displayed on a file that permits the user readily to jump to a desired location in the alphabetical index. Alternatively, the user may search the alphabetical index to find titles or keywords that correspond with an entered character string. A user may select a title in the index and view the file from which the title was extracted.
SUMM . . . files in the file system; a file handler to access the files in response to the scan initiator; and a **header parser** to extract the index-organizing element from the files. In one preferred implementation, the **header parser** terminates access to a file if the formatting scheme of the file is not the same formatting scheme as at least one formatting scheme in a predetermined database of formatting schemes. In another preferred implementation, the **header parser** terminates access to the file if a post-header HTML tag is found before the index-organizing element is found. In yet another preferred implementation, the **header parser** terminates access to the file if more than a predetermined number of bytes of data are encountered before the index-organizing. . .
DETD . . . has successfully accessed a first file from file system 165 as stated above, such first file is then processed by **header parser** 330. File handler 320 also provides to data structure generator 234 the location of such first file as described below. . . generator 234 will expect to encounter it. Each file in file system 165 is passed by file handler 320 to **header parser** 330 in accordance with the operations described above until all such files have been so processed.
DETD **Header parser** 330 opens and reads the first file accessed by file handler 320. Typically, **header parser** 330 will not read the entire file. Rather, **header parser** 330 will examine the contents of the file as they are received to determine if a header is present. A. . .
DETD . . . necessary in the illustrated embodiment that keyword and title format identifier 224 be able to identify the format so that **header parser** 330 may identify the keywords and titles of the files that it examines. It will further be understood that, in. . . files to be examined might not contain headers or might have keywords and titles located in portions other than the **header**. It is advantageous to **group** information such as is typically

contained in a header near the top of a file so that the entire file. . .

DETD **Header parser** 330 will terminate the reading of such
first file if header information is not found in its anticipated
location. Specifically,. . . of formatting used in the files under
examination. In the illustrated embodiment, keyword and title format
identifier 224, having provided **header parser** 330
with formatting information appropriate to the HTML format used in file
system 165, may thus also provide **header parser** 330
with corresponding syntax information regarding the placement of header
information in an HTML-formatted file. Therefore, if **header
parser** 330 encounters a post-**header** tag such as <BODY>
before identifying a header, such condition is indicative of a lack of a
header in the file being read. **Header parser** 330
will in such an instance cease reading the first file. **Header
parser** 330 similarly will cease reading the first file if more
than a pre-determined number of bytes, for example 10,000 bytes,. . .
before encountering a title and one or more keywords as identified by
their formatting as described above. In either case, **header
parser** 330 will return control to file handler 320 so that the
next file is accessed and processed in the manner described above.